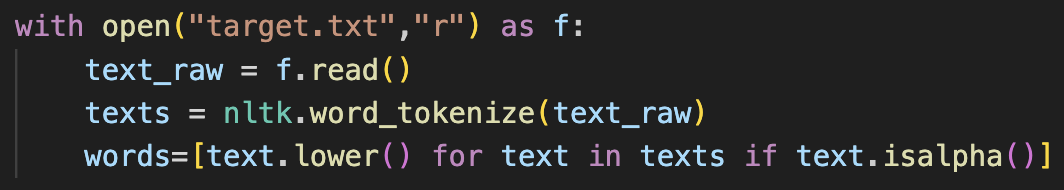
**人工智能程序设计实验报告**

**姓名：孙前普 学号：2023214320**

1. **给定一个文件 target，请使用 matplotlib 可视化工具绘制一个 target 的词云图，如下图所示（可配合使用 wordcloud）：**

**设计思路：**

**1首先使用nltk文字处理库进行统计词频，并且去除标点符号。**

****

**2使用WorldCloud设计词云，wc.generate(words)产生词云，输出图像到目录，得到target.png**

****

**3使用plt.show()将词云现实到屏幕上**

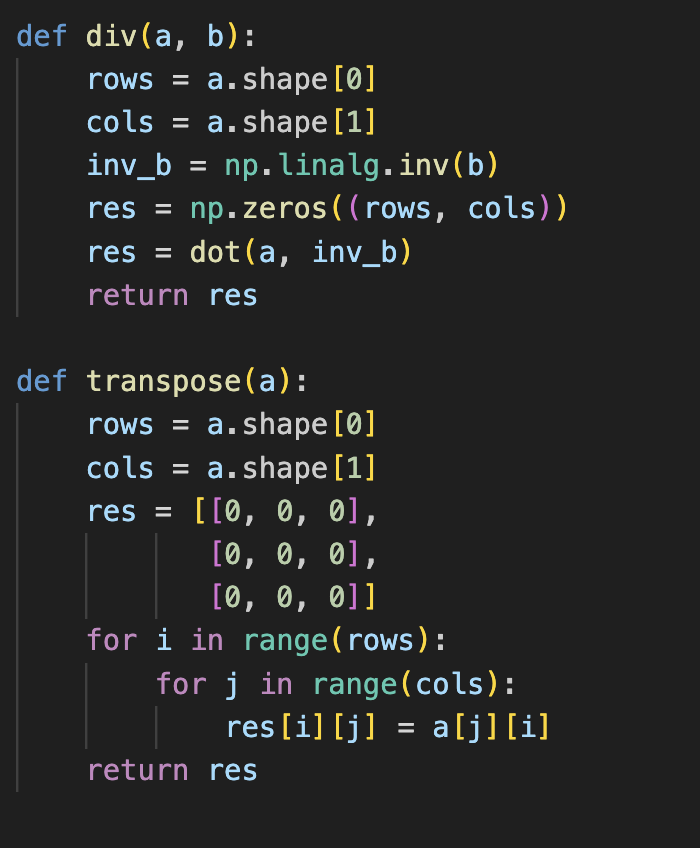
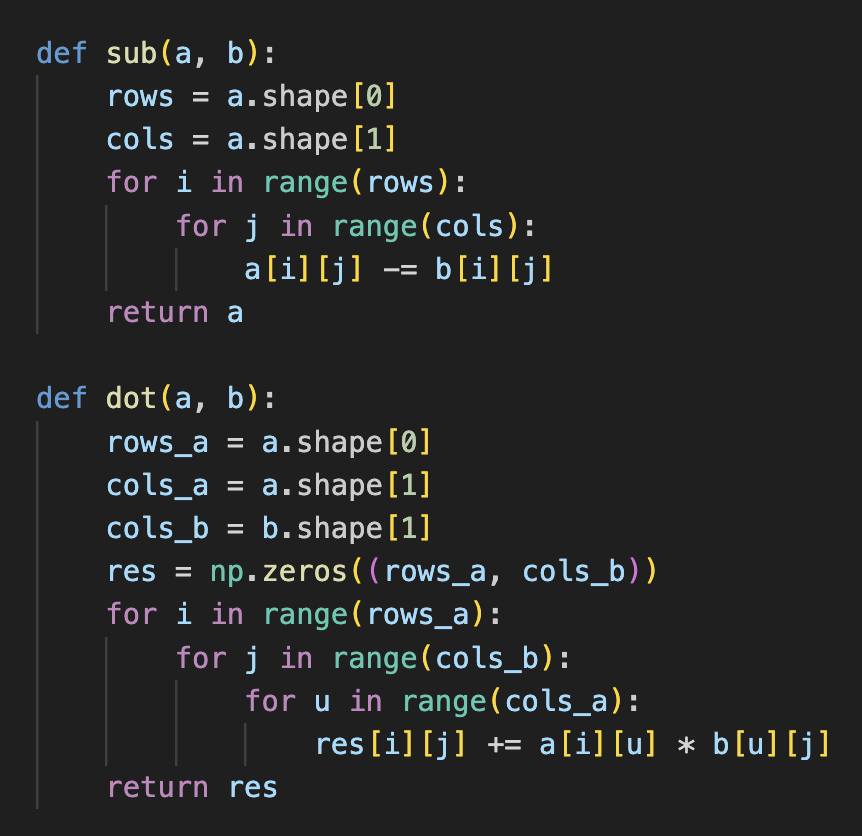
****

1. **在使用 numpy 和不使用 numpy 的情况下，分别实现矩阵的加、减、乘、除、转置。使用时间模块对比运行速度（需要满足任意形状的矩阵运算；除法只考虑方阵，相当于乘逆矩阵）。**

**设计思路：**

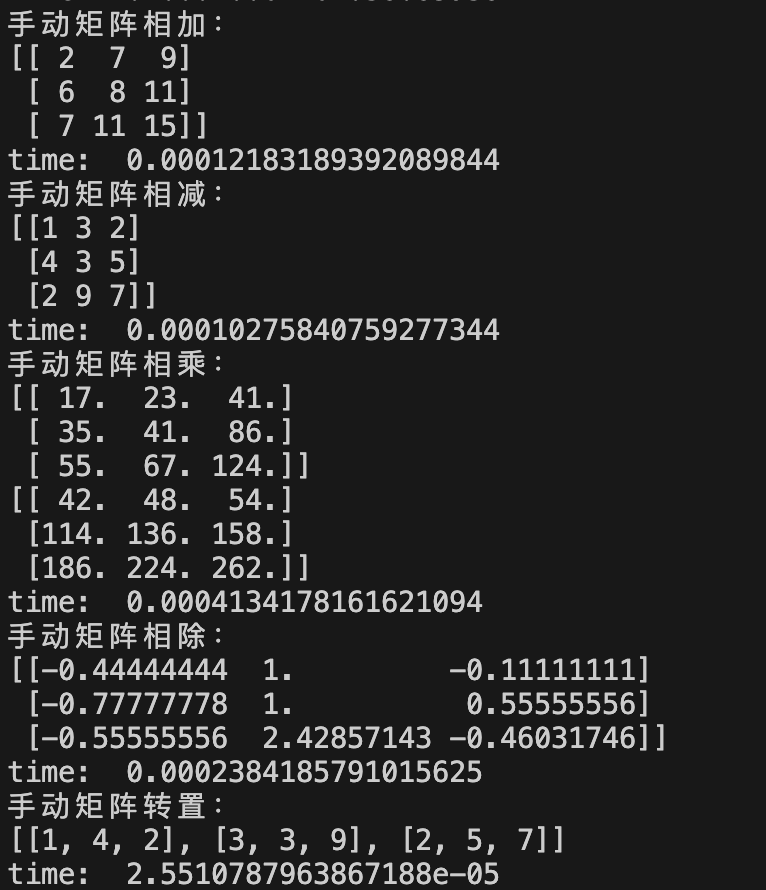
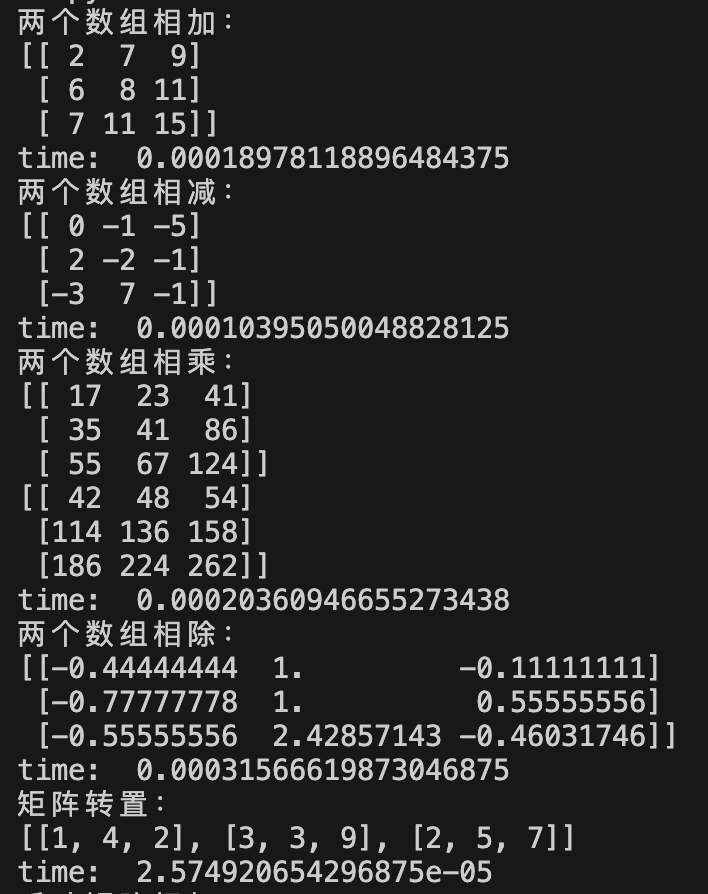
**首先使用numpy进行矩阵的加、减、乘、除、转置计算，得到时间。**

**再设计手动的算法实现，对于加减计算，直接用两个for循环进行遍历矩阵中的对应元素，对于矩阵乘法，使用三个for循环，前两个for循环进行遍历最后的结果矩阵，最后一个for循环遍历a，b两个数组中的对应值，进行相乘再相加，放到res矩阵中。下面是代码实现细节。**

****

**最后计算时间，和numpy的时间进行比较，可以发现对于矩阵加减，numpy和手动的结算结果的时间接近一致，二对于矩阵乘法，手动的计算结果是numpy计算结果的两倍。二转置和除法的计算时间也几乎一致。**

**注意：这里手动除法的矩阵求逆运算使用的也是numpy的直接求逆，而并非手动求逆。**

****

Numpy速度比较快的原因numpy本身的矩阵运算(array operations)可以绕过GIL因为numpy内部是用C写的，不经过python解释器，因此它本身的矩阵运算(array operations)都可以使用多核，此外它内部还用了BLAS(the Basic Linear Algebra Subroutines)，因此可以进一步优化计算速度。